



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

MASANARI TODA

Application No.: 09/779,642

Filed: February 9, 2001

For: INFORMATION PROCESSING
 APPARATUS, INFORMATION
 PROCESSING METHOD, AND
 MEMORY MEDIUM STORING
 PRINT CONTROL PROGRAM
 THEREIN

Group Art Unit: 2185

May 14, 2001

Commissioner for Patents
 Washington, D.C. 20231

CLAIM TO PRIORITY

Sir:

Applicant hereby claims priority under the
 International Convention and all rights to which he is
 entitled under 35 U.S.C. § 119 based upon Japanese Priority
 Application No. 2000-033757, filed February 10, 2000.

A certified copy of the priority document is
 enclosed.

Applicant's undersigned attorney may be reached in
 our Costa Mesa office by telephone at (714) 540-8700. All
 correspondence should continue to be directed to our address
 given below.

Respectfully submitted,

Attorney for Applicant

Registration No. 36,171

FITZPATRICK, CELLA, HARPER & SCINTO
 30 Rockefeller Plaza
 New York, New York 10112-3801
 Facsimile: (212) 218-2200

RECEIVED
 MAY 16 2001
 Group 2100



日本国特許庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application:

2000年 2月10日

出願番号

Application Number:

特願2000-033757

出願人

Applicant (s):

キヤノン株式会社

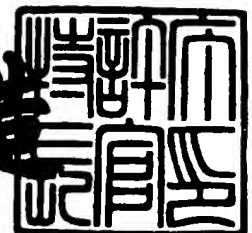
RECEIVED
MAY 16 2001
Group 2100

CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年 3月 2日

特許庁長官
Commissioner,
Patent Office

及川耕造



【書類名】 特許願

【整理番号】 4055028

【提出日】 平成12年 2月10日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 13/00

【発明の名称】 情報処理装置および情報処理方法およびプリンタドライ
バプログラムが格納された記憶媒体

【請求項の数】 20

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会
社内

【氏名】 戸田 雅成

【特許出願人】

【識別番号】 000001007

【氏名又は名称】 キヤノン株式会社

【代理人】

【識別番号】 100077481

【弁理士】

【氏名又は名称】 谷 義一

【選任した代理人】

【識別番号】 100088915

【弁理士】

【氏名又は名称】 阿部 和夫

【手数料の表示】

【予納台帳番号】 013424

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1
【包括委任状番号】 9703598
【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置および情報処理方法およびプリンタドライバプログラムが格納された記憶媒体

【特許請求の範囲】

【請求項 1】 入力された印刷指令情報を格納する記憶手段と、

前記記憶手段に前記印刷指令情報を格納する際に、既に格納されている印刷指令情報の属性が前記印刷指令情報の属性と同一であって且つ前記記憶手段に格納可能な記憶領域がある場合には、既に格納されている印刷指令情報と前記印刷指令情報を合成して前記記憶手段に格納させる合成手段と

を具備したことを特徴とする情報処理装置。

【請求項 2】 請求項 1 に記載の情報処理装置において、

前記合成手段により、前記印刷指令情報と、既に格納されている印刷指令情報とが合成されない場合には、前記記憶手段に格納されている印刷指令情報から中間言語を作成し、その後に、新たな印刷指令情報を前記記憶手段に格納することを特徴とする情報処理装置。

【請求項 3】 請求項 1 に記載の情報処理装置において、

前記印刷指令情報は所定のアプリケーションプログラムを実行させることにより供給され、前記印刷指令情報に基づき 1 ページ分の中間言語を保持してから、印刷装置に対する印刷命令の生成を行うことを特徴とする情報処理装置。

【請求項 4】 請求項 3 に記載の情報処理装置において、

前記印刷命令は、所定の通信媒体を介して前記印刷装置に伝送されることを特徴とする情報処理装置。

【請求項 5】 請求項 1 に記載の情報処理装置において、

前記合成手段は、同一の属性を持つ印刷指令情報を共通ヘッダにまとめることにより、ヘッダサイズ分の削減を行うことを特徴とする情報処理装置。

【請求項 6】 請求項 1 に記載の情報処理装置において、

前記合成手段は、前回保持した描画オブジェクトと現在処理している描画オブジェクトの描画座標変化量を求める手段と、

前記変化量が前回と同じ場合には繰り返し変数をカウントアップする手段とを

備え、

前記変化量が前回と異なる場合には繰り返し変数を表すカウント数と変化量を示すコマンドを前記記憶手段に格納し、それに続いて、現在処理している描画オブジェクトの絶対座標を示すコマンドを前記記憶手段に格納することを特徴とする情報処理装置。

【請求項 7】 入力された印刷指令情報をメモリに格納する際に、既に格納されている印刷指令情報の属性が前記印刷指令情報の属性と同一であって且つ前記記憶手段に格納可能な記憶領域がある場合には、既に格納されている印刷指令情報と前記印刷指令情報を合成して前記メモリに格納させる合成ステップを有することを特徴とする情報処理方法。

【請求項 8】 請求項 7 に記載の情報処理方法において、

前記合成ステップにより、前記印刷指令情報と、既に格納されている印刷指令情報とが合成されない場合には、前記メモリに格納されている印刷指令情報から中間言語を作成し、その後に、新たな印刷指令情報を前記メモリに格納することを特徴とする情報処理方法。

【請求項 9】 請求項 7 に記載の情報処理方法において、

前記印刷指令情報は所定のアプリケーションプログラムを実行させることにより供給され、前記印刷指令情報に基づき 1 ページ分の中間言語を保持してから、印刷装置に対する印刷命令の生成を行うことを特徴とする情報処理方法。

【請求項 10】 請求項 9 に記載の情報処理方法において、

前記印刷命令は、所定の通信媒体を介して前記印刷装置に伝送されることを特徴とする情報処理方法。

【請求項 11】 請求項 7 に記載の情報処理方法において、

前記合成ステップは、同一の属性を持つ印刷指令情報を共通ヘッダにまとめることにより、ヘッダサイズ分の削減を行うことを特徴とする情報処理方法。

【請求項 12】 請求項 7 に記載の情報処理方法において、

前記合成ステップは、前回保持した描画オブジェクトと現在処理している描画オブジェクトの描画座標変化量を求めるステップと、

前記変化量が前回と同じ場合には繰り返し変数をカウントアップするステップ

とを備え、

前記変化量が前回と異なる場合には繰り返し変数を表すカウント数と変化量を示すコマンドを前記メモリに格納し、それに続いて、現在処理している描画オブジェクトの絶対座標を示すコマンドを前記メモリに格納することを特徴とする情報処理方法。

【請求項 1 3】 入力された印刷指令情報をメモリに格納する際に、既に格納されている印刷指令情報の属性が前記印刷指令情報の属性と同一であって且つ前記記憶手段に格納可能な記憶領域がある場合には、既に格納されている印刷指令情報と前記印刷指令情報を合成して前記メモリに格納させる合成ステップを、読み出し可能なプログラムの形態で記憶したことを特徴とする記憶媒体。

【請求項 1 4】 請求項 1 3 に記載の記憶媒体において、

前記合成ステップにより、前記印刷指令情報と、既に格納されている印刷指令情報とが合成されない場合には、前記メモリに格納されている印刷指令情報から中間言語を作成し、その後に、新たな印刷指令情報を前記メモリに格納することを特徴とする記憶媒体。

【請求項 1 5】 請求項 1 3 に記載の記憶媒体において、

前記印刷指令情報は所定のアプリケーションプログラムを実行させることにより供給され、前記印刷指令情報に基づき 1 ページ分の中間言語を保持してから、印刷装置に対する印刷命令の生成を行うことを特徴とする記憶媒体。

【請求項 1 6】 請求項 1 5 に記載の記憶媒体において、

前記印刷命令は、所定の通信媒体を介して前記印刷装置に伝送されることを特徴とする記憶媒体。

【請求項 1 7】 請求項 1 3 に記載の記憶媒体において、

前記合成ステップは、同一の属性を持つ印刷指令情報を共通ヘッダにまとめることにより、ヘッダサイズ分の削減を行うことを特徴とする記憶媒体。

【請求項 1 8】 請求項 1 3 に記載の記憶媒体において、

前記合成ステップは、前回保持した描画オブジェクトと現在処理している描画オブジェクトの描画座標変化量を求めるステップと、

前記変化量が前回と同じ場合には繰り返し変数をカウントアップするステップ

とを備え、

前記変化量が前回と異なる場合には繰り返し変数を表すカウント数と変化量を示すコマンドを前記メモリに格納し、それに続いて、現在処理している描画オブジェクトの絶対座標を示すコマンドを前記メモリに格納することを特徴とする記憶媒体。

【請求項 19】 請求項 13～18 のいずれかに記載の記憶媒体において、前記記憶媒体として、サーバ・コンピュータおよびクライアント・コンピュータが読むことができるプログラムを格納したフロッピーディスク、ハードディスク、光磁気ディスク、光ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMを用いることを特徴とする記憶媒体。

【請求項 20】 請求項 13～18 のいずれかに記載の記憶媒体において、前記記憶媒体は、サーバ・コンピュータおよびクライアント・コンピュータに着脱可能であることを特徴とする記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、情報処理装置および情報処理方法およびプリンタドライバプログラムが格納された記憶媒体に関するものである。

【0002】

更に詳述すると、本発明は、例えば所定の通信媒体を介して印刷装置を制御するのに好適な、ホストコンピュータとして機能する情報処理装置および情報処理方法およびプリンタドライバプログラムが格納された記憶媒体に関するものである。

【0003】

【従来の技術】

まず、従来の印刷システムにおける情報処理方法および処理の流れについて、図 2 を用いて説明する。

【0004】

従来の印刷システムでは、一般には、図 2 に示すように、セントロニクスイン

ターフェースといったパラレル通信手段やネットワーク通信手段を介してホストコンピュータとレーザービームプリンタとが接続されている。

【 0 0 0 5 】

ホストコンピュータ側では、ワードプロセッサや表計算のようなアプリケーションソフトウェア 2 0 1 がウインドウズ (Windows : 米国 Microsoft 社の登録商標) のようないわゆる基本ソフトの上で動作している。こうしたアプリケーションに基づいて印刷を行う場合は、基本ソフトが提供するいくつかのサブシステムのうちグラフィックサブシステム 2 0 2 の機能が用いられる。このグラフィックサブシステムは、例えば Windows では、G D I (Graphic Device Interface) 2 0 2 1 と呼ばれており、ディスプレイあるいはプリンタに対する画像情報の処理を司っている。この G D I 2 0 2 1 は、ディスプレイあるいはプリンタといった各デバイス毎の依存性を吸収するためにデバイスドライバと呼ばれるモジュールを動的にリンクし、それぞれのデバイスに対する出力処理を行う。プリンタに対するこのモジュールは、プリンタドライバ 2 0 2 3 と呼ばれる。

【 0 0 0 6 】

このプリンタドライバ 2 0 2 3 は、その能力や機能などに応じて予めデバイスドライバに実装することが決められている D D I 2 0 2 2 (Device Driver Interface) と呼ばれる関数群を用意する必要がある。アプリケーションの A P I (Application Programming Interface) コールについては、G D I 2 0 2 1 がデバイスドライバ用にデータ変換し、上記 D D I 関数群 2 0 2 2 が適宜 G D I 2 0 2 1 からコールされて所定の印刷処理が実行されるようになっている。G D I 2 0 2 1 は、このようにプリンタドライバ 2 0 2 3 を介してアプリケーションからの印刷要求をシーケンシャルに処理している。一般的には、このように O S の枠組み内にあるプリンタドライバ 2 0 2 3 内で P D L コマンドを生成したり、イメージモードとしてレンダリング処理を行っている。

【 0 0 0 7 】

また、イメージモード時のバンディング処理を独自で行ったり、逆順印刷等のページハンドリング処理・処理モードの自動切り替えといった O S がサポートしていない機能を実現するために、プリンタドライバ 2 0 2 3 は受け取った描画命

令から中間言語ファイル（PDF 2 0 3 1）の生成のみを行い、アプリケーションであるスプールサブシステム 2 0 3 のデスプーラ 2 0 3 2 が PDF 2 0 3 1 を読み込み、付加機能の追加を行っている。

【 0 0 0 8 】

プリンタドライバの処理は、PDLモード（2 0 3 3 参照）とイメージモード（2 0 3 4 ～ 2 0 3 6 参照）の 2 種類に大別できる。

【 0 0 0 9 】

PDLモード 2 0 3 3 は、印刷装置側が PDL（Printer Description Language）と呼ばれる制御コマンドを処理できるコントローラを搭載している場合に用いられ、PDF 2 0 3 1 から PDL のコマンドに変換し次第、インタフェース（I/F）部 2 0 3 8 を介して印刷装置（プリンタ）1 0 0 に送出する。

【 0 0 1 0 】

PDLモード時の処理は、DDI 2 0 2 2 で受け取ったデータをシーケンシャルにコマンド生成するだけであるので、ドライバにとっては軽い処理といえる。PDLモードタイプでは、プリンタの PDL コマンドの解析能力が高ければ GDI 2 0 2 1 から受け取った抽象度の高いデータをそのまま送出できるので、文字中心の通常ページ印刷については出力階調に拘わらず高速である、という特徴をもつ。

【 0 0 1 1 】

しかし、グラフィックス系アプリケーションで複雑な図形・高解像度・高階調イメージを出力しようとする、と、PDL コマンドが大きくなり転送速度・プリンタの処理速度が落ちる。理論的には PDL コマンドサイズに上限はない。

【 0 0 1 2 】

一方、イメージモードは、プリンタドライバ側で確保したバンドメモリ空間上に印刷イメージの展開を行い、それを印刷装置で直接印刷できる形態に変換し、印刷装置に送り印刷するものである。

【 0 0 1 3 】

イメージモードタイプでは、転送データサイズが描画面積と出力階調数で決まるため、簡単なページでも複雑なページでも転送速度・プリンタの処理時間は変

わない、という特徴がある。そのため、PDLモードで処理時間のかかるページでも比較的高速な印刷を行うことが可能であるが、文字だけの簡単なページでもプリンタをエンジンスルーで動かすことは難しい。

【0014】

デスプーラ2032は、PDF2031をデスプーリングする際に、適当な高さに分割したバンドメモリを用い、複数回に分けて描画位置に関する描画命令をイメージモード用モジュール2034～2036に渡して描画処理を行う。生成したビットマップデータは、バンド毎にI/F部2038を介して印刷装置100に送出される。

【0015】

イメージモードの特徴として、高解像度カラーイメージや複雑なグラフィックデータの描画処理が高速である点が挙げられる。しかし、文字あるいは簡単な図形についても、描画領域全体のイメージデータを出力解像度・出力階調で生成し、印刷命令として送出するので、高解像度・高階調ページの高速印刷には適していない。

【0016】

これらの両モードとも、描画処理を行うために、イメージモードではホストコンピュータ側に、PDLモードではプリンタ側に1ページ分の描画用メモリ領域が必要である。ところが、カラープリンタではRGB各色8ビットで600DPI、A4フルページ分のメモリを確保しようとするすると96メガバイトが必要であり、現状ではホストマシンでも確保することは困難である。そこで、カラープリンタの描画処理では、出力形式であるYMCK各色1, 2または4ビットの色空間を使い、さらにページをいくつかの領域に分割したバンドメモリを使うことで、描画処理用メモリの削減を行っている。

【0017】

このように、1ページを複数のバンドに分割して描画処理を行うことをバンディング処理と呼んでいる。このバンディング処理は、PDLモードの場合はプリンタ側で、イメージモードの場合はホストマシン側で1ページ分の描画命令を保持し、各バンドに必要な印刷命令のみを処理することで、各バンドの描画処理を

実現している。

【0018】

【発明が解決しようとする課題】

しかし、上記のDDI2022を介して渡される描画命令の中には、1ライン描画関数(Windows9X;Output 関数のSCANLINE)、数ピクセルの矩形塗りつぶし関数(Windows9X;Output 関数のRECTANGLE)が何度もコールされて、複雑な形の図形描画処理を行う場合がある。こういった場合に、プリンタドライバで1ページ分のスプール処理を行うと、中間言語数・データサイズの増大が起こり、速度低下の要因となってしまうという問題が生じる。

【0019】

よって本発明の目的は、上述の点に鑑み、プリンタドライバで中間言語を一旦スプールする場合でも、保持データサイズ・個数を抑えることで高速な印刷を可能とした、情報処理装置および情報処理方法を提供することにある。

【0020】

【課題を解決するための手段】

上記の目的を達成するために、請求項1に係る本発明は、入力された印刷指令情報を格納する記憶手段と、前記記憶手段に前記印刷指令情報を格納する際に、既に格納されている印刷指令情報の属性が前記印刷指令情報の属性と同一であって且つ前記記憶手段に格納可能な記憶領域がある場合には、既に格納されている印刷指令情報と前記印刷指令情報を合成して前記記憶手段に格納させる合成手段とを具備した情報処理装置である。

【0021】

請求項2に係る本発明は、請求項1に係る情報処理装置において、前記合成手段により、前記印刷指令情報と、既に格納されている印刷指令情報とが合成されない場合には、前記記憶手段に格納されている印刷指令情報から中間言語を作成し、その後、新たな印刷指令情報を前記記憶手段に格納する。

【0022】

請求項3に係る本発明は、請求項1に係る情報処理装置において、前記印刷指令情報は所定のアプリケーションプログラムを実行させることにより供給され、

前記印刷指令情報に基づき 1 ページ分の中間言語を保持してから、印刷装置に対する印刷命令の生成を行う。

【 0 0 2 3 】

請求項 4 に係る本発明は、請求項 3 に係る情報処理装置において、前記印刷命令は、所定の通信媒体を介して前記印刷装置に伝送される。

【 0 0 2 4 】

請求項 5 に係る本発明は、請求項 1 に係る情報処理装置において、前記合成手段は、同一の属性を持つ印刷指令情報を共通ヘッダにまとめることにより、ヘッダサイズ分の削減を行う。

【 0 0 2 5 】

請求項 6 に係る本発明は、請求項 1 に係る情報処理装置において、前記合成手段は、前回保持した描画オブジェクトと現在処理している描画オブジェクトの描画座標変化量を求める手段と、前記変化量が前回と同じ場合には繰り返し変数をカウントアップする手段とを備え、前記変化量が前回と異なる場合には繰り返し変数を表すカウント数と変化量を示すコマンドを前記記憶手段に格納し、それに続いて、現在処理している描画オブジェクトの絶対座標を示すコマンドを前記記憶手段に格納する。

【 0 0 2 6 】

請求項 7 に係る本発明は、入力された印刷指令情報をメモリに格納する際に、既に格納されている印刷指令情報の属性が前記印刷指令情報の属性と同一であって且つ前記記憶手段に格納可能な記憶領域がある場合には、既に格納されている印刷指令情報と前記印刷指令情報を合成して前記メモリに格納させる合成ステップを有する情報処理方法である。

【 0 0 2 7 】

請求項 8 に係る本発明は、請求項 7 に係る情報処理方法において、前記合成ステップにより、前記印刷指令情報と、既に格納されている印刷指令情報とが合成されない場合には、前記メモリに格納されている印刷指令情報から中間言語を作成し、その後に、新たな印刷指令情報を前記メモリに格納する。

【 0 0 2 8 】

請求項 9 に係る本発明は、請求項 7 に係る情報処理方法において、前記印刷指令情報は所定のアプリケーションプログラムを実行させることにより供給され、前記印刷指令情報に基づき 1 ページ分の中間言語を保持してから、印刷装置に対する印刷命令の生成を行う。

【 0 0 2 9 】

請求項 1 0 に係る本発明は、請求項 9 に係る情報処理方法において、前記印刷命令は、所定の通信媒体を介して前記印刷装置に伝送される。

【 0 0 3 0 】

請求項 1 1 に係る本発明は、請求項 7 に係る情報処理方法において、前記合成ステップは、同一の属性を持つ印刷指令情報を共通ヘッダにまとめることにより、ヘッダサイズ分の削減を行う。

【 0 0 3 1 】

請求項 1 2 に係る本発明は、請求項 7 に係る情報処理方法において、前記合成ステップは、前回保持した描画オブジェクトと現在処理している描画オブジェクトの描画座標変化量を求めるステップと、前記変化量が前回と同じ場合には繰り返し変数をカウントアップするステップとを備え、前記変化量が前回と異なる場合には繰り返し変数を表すカウント数と変化量を示すコマンドを前記メモリに格納し、それに続いて、現在処理している描画オブジェクトの絶対座標を示すコマンドを前記メモリに格納する。

【 0 0 3 2 】

請求項 1 3 に係る本発明は、入力された印刷指令情報をメモリに格納する際に、既に格納されている印刷指令情報の属性が前記印刷指令情報の属性と同一であって且つ前記記憶手段に格納可能な記憶領域がある場合には、既に格納されている印刷指令情報と前記印刷指令情報を合成して前記メモリに格納させる合成ステップを、読み出し可能なプログラムの形態で記憶した記憶媒体である。

【 0 0 3 3 】

請求項 1 4 に係る本発明は、請求項 1 3 に係る記憶媒体において、前記合成ステップにより、前記印刷指令情報と、既に格納されている印刷指令情報とが合成されない場合には、前記メモリに格納されている印刷指令情報から中間言語を作

成し、その後、新たな印刷指令情報を前記メモリに格納する。

【 0 0 3 4 】

請求項 1 5 に係る本発明は、請求項 1 3 に係る記憶媒体において、前記印刷指令情報は所定のアプリケーションプログラムを実行させることにより供給され、前記印刷指令情報に基づき 1 ページ分の中間言語を保持してから、印刷装置に対する印刷命令の生成を行う。

【 0 0 3 5 】

請求項 1 6 に係る本発明は、請求項 1 5 に係る記憶媒体において、前記印刷命令は、所定の通信媒体を介して前記印刷装置に伝送される。

【 0 0 3 6 】

請求項 1 7 に係る本発明は、請求項 1 3 に係る記憶媒体において、前記合成ステップは、同一の属性を持つ印刷指令情報を共通ヘッダにまとめることにより、ヘッダサイズ分の削減を行う。

【 0 0 3 7 】

請求項 1 8 に係る本発明は、請求項 1 3 に係る記憶媒体において、前記合成ステップは、前回保持した描画オブジェクトと現在処理している描画オブジェクトの描画座標変化量を求めるステップと、前記変化量が前回と同じ場合には繰り返し変数をカウントアップするステップとを備え、前記変化量が前回と異なる場合には繰り返し変数を表すカウント数と変化量を示すコマンドを前記メモリに格納し、それに続いて、現在処理している描画オブジェクトの絶対座標を示すコマンドを前記メモリに格納する。

【 0 0 3 8 】

請求項 1 9 に係る本発明は、請求項 1 3 ～ 1 8 のいずれかに係る記憶媒体において、前記記憶媒体として、サーバコンピュータおよびクライアントコンピュータが読むことができるプログラムを格納したフロッピーディスク、ハードディスク、光磁気ディスク、光ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMを用いる。

【 0 0 3 9 】

請求項 2 0 に係る本発明は、請求項 1 3 ～ 1 8 のいずれかに係る記憶媒体にお

いて、前記記憶媒体は、サーバコンピュータおよびクライアントコンピュータに着脱可能である。

【0040】

【発明の実施の形態】

以下、図面を参照して、本発明の実施の形態を詳細に説明する。

【0041】

図1は、本発明を適用した印刷システムに含まれるカラーレーザービームプリンタ（以下「カラーLB P」と記す）の断面構成を示す。すなわち本図は、600ドット／インチ（dpi）の解像度を有し、各色成分各画素が8ビットで表現された多値データに基づいて画像記録を行うカラーLB Pの構造を示す側断面図である。

【0042】

図1において、100はカラーLB P本体（以下、印刷装置ともいう）であり、外部に接続されているホストコンピュータなどから供給されるプリントデータ（文字コードや画像データ等）および制御コードから成る印刷情報を入力して記憶すると共に、それらの情報に従って対応する文字パターンやイメージ等を作成し、記憶媒体である記録紙上に像を形成する装置である。

【0043】

110はホストコンピュータから供給される印刷情報を解析し印刷イメージの生成処理を行うと共に、カラーLB P本体100の制御を行うフォーマッタ制御部である。このフォーマッタ制御部110は、ユーザによる操作およびユーザに対する状態通知のためのスイッチおよびLED表示器等が配されているオペレーションパネル部120と接続されており、そのパネル部は印刷装置100の外装の一部として配設されている。フォーマッタ制御部110において生成された最終的な印刷イメージはビデオ信号として出力制御部130に送出される。出力制御部130は、印刷装置100の不図示の各種センサからの状態入力と共に光学ユニット140および各種駆動系機構部に対して制御信号を出力し、印刷装置100としての印刷処理の制御を司る。

【0044】

図 1 に示す印刷装置において、給紙カセット 1 6 1 から給紙された用紙（転写紙）P はその先端をグリッパ 1 5 4 f により挟持されて、転写ドラム 1 5 4 の外周に保持される。光学ユニット 1 4 0 により感光ドラム 1 5 1 上に形成された各色の潜像は、イエロー（Y）、マゼンタ（M）、シアン（C）、ブラック（B）の各色現像器 D y、D m、D c、D b により現像化されて、転写ドラム外周の用紙に複数回転写され、多色画像が形成される。その後、用紙 P は転写ドラム 1 5 4 より分離されて、定着ユニット 1 5 5 で定着され、排紙部 1 5 9 より排紙トレイ 1 6 0 に排出される。

【 0 0 4 5 】

各色の現像器 D y、D m、D c、D b は、その両端に回転支軸を有し、各々がその軸を中心に回転可能となるように現像器選択機構部 1 5 2 により保持される。これによって、各現像器 D y、D c、D b、D n は、図 1 に示すように現像器選択のために現像器選択機構部 1 5 2 が回転軸 1 5 2 a を中心にして回転しても、その姿勢を一定に維持できる構成をとっている。選択された現像器が現像位置に移動後、支点 1 5 3 b を中心にして、選択機構保持フレーム 1 5 3 がソレノイド 1 5 3 a により感光ドラム 1 5 1 方向へ引っ張られ、感光ドラム 1 5 1 方向へ移動して現像処理が行われる。次に、帯電器 1 5 6 によって感光ドラム 1 5 1 が所定の極性に均一に帯電される。フォーマッタ制御部 1 1 0 において画像イメージとして展開された印刷情報は、対応するパターンのビデオ信号に変換され、レーザドライバを介して半導体レーザ 1 4 1 を駆動する。入力されたビデオ信号に応じて半導体レーザ 1 4 1 から発射されるレーザ光はオンオフ制御され、さらにスキャナモータ 1 4 3 によって高速回転するポリゴンミラー 1 4 2 で左右方向に振らされ、ポリゴンレンズ 1 4 4、反射鏡 1 4 5 を介して感光ドラム 1 5 1 上を走査露光する。これにより、感光ドラム 1 5 1 上には画像パターンの静電潜像が形成されることになる。次に、例えば、M（マゼンタ）色の静電潜像が M（マゼンタ）色の現像器 D m により現像され、感光体ドラム 1 5 1 上に M（マゼンタ）色の第 1 のトナー像が形成される。

【 0 0 4 6 】

一方、所定のタイミングで転写紙 P が給紙され、トナーと反対極性（例えばブ

ラス極性)の転写バイアス電圧が転写ドラム154に印加され、感光体ドラム151上の第1トナー像が転写紙Pに転写されると共に、転写紙Pが転写ドラム154の表面に静電吸着される。その後、感光ドラム151はクリーナー157によって残留するM(マゼンタ)色トナーが除去され、次の色の潜像形成および現像行程に備える。

【0047】

以下同様の手順によってC(シアン), Y(イエロー), Bk(ブラック)の順で第2, 3, 4色目のトナー像の転写が行われる。但し、各色の転写時には、転写ドラム154には前回よりも高いバイアス電圧が印加される点が異なる。4色のトナー像が重畳転写された転写紙Pの先端部が分離位置に近づくと、分離爪158が接近してその先端が転写ドラム154の表面に接触し、転写紙Pを転写ドラム154から分離させる。分離された転写紙Pは定着ユニット155に搬送され、ここで転写紙上のトナー像が定着されて排紙トレイ160上に排出される。

【0048】

本実施の形態におけるカラーレーザビームプリンタは、以上のような画像形成過程を経て600ドット/インチ(dpi)の解像度で画像出力を行う。なお、本発明を適用可能なプリンタは、カラーLBPに限られるものではなく、インクジェットプリンタやサーマルプリンタ等、他のプリント方式のカラープリンタでもよい。

【0049】

次に、カラーLBP(印刷装置)100におけるフォーマッタ制御部110について図3を用いて説明する。このフォーマッタ制御部110は、通常はPDLコントローラなどとも呼ばれており、ホストコンピュータとの接続手段であるインターフェース(I/F)部111と、受信データ等を一時的に保持管理するための受信バッファ1121と、送信データ等を一時的に保持管理するための送信バッファ1122と、印刷データの解析を司るコマンド解析部113と、印刷制御処理実行部114と、描画処理実行部115と、ページメモリ116とを含んでいる。

【 0 0 5 0 】

インターフェース（I/F）部 1 1 1 は、ホストコンピュータ 2 0 0 との印刷データの送受信を行う通信手段あり、通信プロトコルとして I E E E - 1 2 8 4 に準拠した通信を可能とするものである。但し、本実施の形態では、この通信手段に限定するものでなく、ネットワークを介してさまざまなプロトコルによる接続であってもよいし、I E E E - 1 3 9 4 に準じた通信手段であってもよい。このインターフェース部 1 1 1 を通して受信した印刷データは、そのデータを一時的に保持する記憶手段である受信バッファ 1 1 2 1 に逐次蓄積され、必要に応じてコマンド解析部 1 1 3 または描画処理実行部 1 1 5 によって読み出され処理される。

【 0 0 5 1 】

コマンド解析部 1 1 3 は、各 P D L コマンド体系や印刷ジョブ制御言語に準じた制御プログラムにより構成されており、文字印字、図形、イメージなどの描画に関する印刷データの解析結果は、描画処理実行部 1 1 5 に指示を与えて処理し、給紙選択やリセット命令などの描画以外のコマンドは、印刷制御処理実行部 1 1 4 に指示を出し処理する。

【 0 0 5 2 】

描画処理実行部 1 1 5 は、文字やイメージの各描画オブジェクトをページメモリ 1 1 6 に逐次展開して行くレンダラである。

【 0 0 5 3 】

図 1 に示したカラー L B P に対しては、M C Y K の面順次でエンジン回転処理に間に合うようにページデータを送出する必要があるが、標準状態では、そのために必要なメモリをすべて確保するわけではなく、1 プレーン（1 または 2 bit/pixel）の数分の 1 のバンド領域としてメモリが確保され、そのバンド領域を使いまわして画像をエンジン速度に同期して処理するように構成してある。通常は、このように Y M C K レンダラによる展開処理とプリンタエンジンへのビデオ信号の SHIPPING との追いかかけ、所謂バンディング制御によってページメモリ 1 1 6 は管理されているが、十分なメモリがある場合は、1 ページ分が展開可能な領域を確保しても良い。

【 0 0 5 4 】

なお、一般的に、フォーマッタ制御部 1 1 0 は、中央演算処理装置（CPU）、リードオンリーメモリ（ROM）、ランダムアクセスメモリ（RAM）などを用いたコンピュータシステムによって構成されている。また、各部の処理は、マルチタスクモニタ（リアルタイムOS）のもとでタイムシェアリングに処理される構成であっても良いし、各機能ごとに専用のコントローラハードウェアを用意して独立して処理される構成であってもかまわない。

【 0 0 5 5 】

オペレーションパネル 1 2 0 は、前述した通り印刷装置の各種状態を設定・表示するためのものである。出力制御部 1 3 0 は、ページメモリ 1 1 6 の内容をビデオ信号に変換処理し、プリンタエンジン部 1 5 0 への画像転送を行う。プリンタエンジン部 1 5 0 は受け取ったビデオ信号を記録紙に永久可視画像形成するための印刷機構部であり、図 1 において前述したものである。

【 0 0 5 6 】

以上、LBP プリンタ（印刷装置） 1 0 0 について説明したが、次にホストコンピュータ 3 0 0 を含む印刷システムの全体構成について説明していく。

【 0 0 5 7 】

図 3 において、3 0 0 はホストコンピュータであり、プリントデータおよび制御コードから成る印刷情報を印刷装置 1 0 0 に出力する。ホストコンピュータ 3 0 0 は、入力デバイスであるキーボード 3 1 0 や、ポインティングデバイスであるマウス 3 1 1 と、表示デバイスであるディスプレイモニタ 3 2 0 を合わせた一つのコンピュータシステムとして構成されている。このホストコンピュータ 3 0 0 は、Windows NT、Windows95(98) などの基本OSによって動作しているものとする。

【 0 0 5 8 】

ホストコンピュータ側について、本実施の形態に関する機能的な部分にのみ注目し、基本OS上での機能を大きく分類すると、アプリケーションソフトウェア 3 0 1、グラフィックサブシステム 3 0 2、印刷情報格納手段および印刷装置との通信手段を含むスプールサブシステム 3 0 3 に大別される。アプリケーション

ソフトウェア 3 0 1 は、例えば、ワープロや表計算などの基本ソフトウェア上で動作する応用ソフトウェアを指すものである。グラフィックサブシステム 3 0 2 は、基本 OS の機能の一部である Graphic Device Interface (以後、G D I と記す) 3 0 2 1 とその G D I から動的にリンクされるデバイスドライバであるところのプリンタドライバ 3 0 2 2 によって構成されている。

【 0 0 5 9 】

プリンタドライバ 3 0 2 3 は、G D I から D D I (Device Driver Interface) 3 0 2 2 というインターフェースを介してコールされ、デバイスに応じた処理を描画オブジェクト毎に行うものである。本システムでは、プリンタドライバが、D D I 関数を介して受け取った描画命令の情報を、描画オブジェクトタイプ毎のタグをつけて、描画位置情報、色情報、パターン情報、描画論理情報といった属性と共に、一次バッファ 3 0 2 4 に保持する。

【 0 0 6 0 】

但し、一次バッファ 3 0 2 4 に入りきらないような描画オブジェクト (巨大なイメージデータ等) は保持しない。

【 0 0 6 1 】

一次バッファ 3 0 2 4 に保持する際に、既に保持されている描画オブジェクトがある場合、タイプ、描画属性が同じ時は一次バッファ 3 0 2 4 の中に追加登録していく。「タイプが異なる。」「一次バッファにデータが入りきらない」、「描画属性が変化する」、「ページ処理終了」の何れかに当てはまるときは、既に一次バッファ 3 0 2 4 に入っている描画オブジェクトから P D F 3 0 3 1 (ページデータファイル) を生成し、スプールする。

【 0 0 6 2 】

スプールサブシステム 3 0 3 は、グラフィックサブシステム 3 0 2 の後段に位置するプリンタデバイスに特有のサブシステムであり、デスプーラ 3 0 3 2 を含んでいる。デスプーラ 3 0 3 2 は、印刷情報格納手段であるところの P D F 3 0 3 1 を読み込み、指定されたモード用トランスレータ (3 0 3 3 ~ 3 0 3 6) でデスプーラ処理を行うものである。選択されたドライバ (3 0 3 3 ~ 3 0 3 6) は、イメージモード (3 0 3 4 ~ 3 0 3 6) であれば、バンドメモリ 3 0 3 7 を

用いてイメージ展開を行い、PDLモード（3033）であればスプールファイルからPDLコマンドへの変換処理を行い、I/F部3038を介してプリンタにデータを送出する。

【0063】

基本OSによって、上述したこれらの名称や機能的な枠組みは若干異なる場合があるが、本実施の形態で言う各技術的手段が実現できるモジュールであれば、それらの名称や枠組みはあまり大きな問題ではない。例えば、スプーラやスプールファイルと呼ばれるものは、別のOSにおいてプリントキューと呼ばれるモジュールに処理を組み込むことによっても実現可能である。

【0064】

なお一般的に、これらの各機能モジュールを含むホストコンピュータ300は、中央演算処理装置（CPU）、リードオンリーメモリ（ROM）、ランダムアクセスメモリ（RAM）、ハードディスクドライブ（HDD）、各種入出力制御部（I/O）などのハードウェアのもとで、基本ソフトと呼ばれるソフトウェアがその制御を司り、その基本ソフトの元で、それぞれの応用ソフト、サブシステムプロセスが機能モジュールとして動作するようになっている。

【0065】

次に、一次バッファを使ったPDFデータ圧縮方法の更に詳しい説明を行う。

【0066】

ここでは、本実施の形態における動作を説明するために、Windows95(Windows98)のDDIの説明をしておく。

【0067】

Windows9XのDDIにはグラフィックス系関数の一つとして、

```
Output(LPPDEVICE lpDestDev, WORD wStyle,
        WORD wCount,  LPPPOINT lpPoints,
        LPPEN lpPPen,  LPPBRUSH lpPBrush,
        LPDRAWMODE    lpDrawMode,
        LPRECT lpClipRect )
```

が用意されている。

【 0 0 6 8 】

本関数は、第二引数のwStyleの値によって、渡されるデータの意味が変化する。例えば、OS_RECTANGLEがセットされていた場合、wCountには必ず2がセットされる。本関数は第4引数のlpPointsが示す最初の座標情報（X，Y）は矩形の左上を、次の座標情報（X，Y）は右下の点を示し、その対角線を持つ矩形の内側をlpPBrushが示す色、パターン、lpPPenが示す枠線で、描画処理することが求められる。

【 0 0 6 9 】

プリンタドライバ3023により、DDI関数毎にPDFを生成すると図10に示す1000のようなデータ構造のファイルが生成される。本図において、1000はヘッダ部1001とデータ部1002で構成されている。ヘッダ部1001の先頭にはデータタイプ1003を持っている。ヘッダ部1001には他にブラシ情報1004、ペン情報1005、データサイズCount,論理演算値1006、外接矩形情報1007等を持っている。データ部1002は、POINT型構造体配列で構成されており、Count 1005数分の配列にデータが格納されている。

【 0 0 7 0 】

しかし、グラフィックス系アプリケーションでグラデーションを描いて出力しようとする、図4の400で示すような細かなRECTANGLEを多量に指定して行く場合がある。すなわち、400のA行-[5～6]列が401、B行-[4～6]列が402、C行-[3～6]列が403・・・である。

【 0 0 7 1 】

このように、DDI関数に渡された細かな単位で矩形描画命令を保持するとPDFのサイズが膨れ上がり、PDLモード、イメージモード共に印字速度の低下に繋がる。そこで、本実施の形態は、プリンタドライバ3023ではDDI関数がコールされても直ちにPDFを生成せずに、固定サイズのメモリ空間である一次バッファ3024を使うことでPDFの圧縮処理を行う。

【 0 0 7 2 】

一次バッファ3024は、ページ内で保持される必要があり、かつドライバが

提供している全ての D D I 関数からもアクセス可能でなくてはならない。Windows 9X の全ての D D I には “LPPDEVICE lpDestDev” が第一引数に入っており、一つのジョブ内で保持されるメモリ空間へのポインタを示している。一次バッファへのポインタはメンバとして登録する。

【 0 0 7 3 】

本実施の形態における圧縮処理は、大きく分けて以下の二つの手法の組み合わせによって実現する。

【 0 0 7 4 】

- 1) ヘッダ部の共通化によるサイズ、コマンド数の削減
- 2) データ部圧縮によるデータサイズの削減

以下、それぞれについての詳しい説明を行う。

【 0 0 7 5 】

まず、「1) ヘッダ部の共通化によるサイズ、コマンド数の削減」について説明する。

【 0 0 7 6 】

細かく分割された同じ属性の描画命令は、ヘッダ情報の共通化で P D F のサイズ削減が可能である。

【 0 0 7 7 】

前回の描画属性と比較するために、以下のようなメンバーが必要である。

【 0 0 7 8 】

```
typedef struct
{
    enum ELEMTYPE lasDataType;    //前回データタイプ
    WORD lastPenIndex;            //前回データのペン情報
    WORD lastBrushIndex;          //前回データのブラシ情報
    DRAWMODE lastDrawMode;        //前回データのDrawMode
    RECT lastboundRect;           //前回データの外接矩形
    RECT BoundRect;               //圧縮データ全体の外接矩形
    LPSTR lpstart;                 //圧縮データの先頭ポインタ
}
```

```

LPSTR  lplast;           //圧縮データの現状の最終ポインタ
WORD    bufsize;        //一次バッファサイズ
} LASTDATA;

```

【 0 0 7 9 】

プリンタドライバが提供する D D I 関数内では、上記LASTDATAにセットされている前回のデータタイプと今回のデータを比較し、lasDataType が同じときは、タイプ別圧縮処理関数をコールし、違う場合は一次バッファ内のデータから P D F の生成を行う。

【 0 0 8 0 】

図 5 は、以上の処理の流れを示すフローチャートである。図 5 における各ステップの処理内容は、以下の通りである。

【 0 0 8 1 】

ステップ S 6 0 1 : 一次バッファ内に描画データが格納されているならステップ S 6 0 6 へ、一次バッファにデータがセットされていない場合はステップ S 6 0 2 へ制御を移す。

【 0 0 8 2 】

ステップ S 6 0 2 : 一次バッファ内に格納した描画データのタイプ、属性（ブラシ、ペン、論理演算種）と同じならばステップ S 6 0 3 へ、違う場合は 6 0 4 へ制御を移す。

【 0 0 8 3 】

ステップ S 6 0 3 : 一次バッファ内に渡されたデータを格納可能ならばステップ S 6 0 5 へ、不可能ならステップ S 6 0 4 へ制御を移す。

【 0 0 8 4 】

ステップ S 6 0 4 : 一次バッファに格納されている描画データから P D F を作製し、ステップ S 6 0 5 へ制御を移す。

【 0 0 8 5 】

ステップ S 6 0 5 : 現在の D D I 関数に渡された描画データをデータタイプ別圧縮関数に渡して一次バッファにセットする。その後、ステップ S 6 0 6 へ制御を移す。

【 0 0 8 6 】

ステップ S 6 0 6 : LASTDATAを更新し、ステップ S 6 0 7 へ制御を移す。

lastDataType	変更があった場合はセット
lastPenIndex	変更があった場合はセット
lastBrushIndex	変更があった場合はセット
lastDrawMode	変更があった場合はセット
lastboundRect	今回の描画命令の外接矩形をセット
BoundRect	前回のBoundRect と今回の外接矩形のAND を取る。
lpstart	一次バッファの先頭アドレス、変更なし
lpplast	一次バッファ内の現在のデータを詰めこんだ最終ポインタ
bufsize	今回詰めこんだデータサイズを足しこむ。

【 0 0 8 7 】

ステップ S 6 0 7 : 描画命令全て終了ならステップ S 6 0 8 へ、まだならステップ S 6 0 1 へ制御を移す。

【 0 0 8 8 】

ステップ S 6 0 8 : 一次バッファ内に残っている描画データから P D F を作製する。

【 0 0 8 9 】

図 6 は、上記のステップ S 6 0 4 またはステップ S 6 0 8 によって得られた結果を例示したものである。

【 0 0 9 0 】

次に、「2) データ部圧縮によるデータサイズの削減」について説明する。

【 0 0 9 1 】

一次バッファ 3 0 2 4 に既に格納されている描画命令と同一タイプの描画命令がコールされた時にコールされるデータタイプ別圧縮関数は、座標空間の連続性等を活かしてデータ部の圧縮処理を行う。

【 0 0 9 2 】

具体例として、Windows9X, Output(Scanline)の圧縮方法の一例を示す。

【 0 0 9 3 】

1. Output(Scanline)の圧縮アルゴリズムについて

図 7 を参照しながらOutput(Scanline)関数の説明を行う。D D I 関数の一つである Output関数の第 2 引数にOS_SCANLINESがセットされているときには、lpPointsが示す先のPOINT 構造体配列は、

lpPoints [0]->y : Y座標

lpPoints [1]->x : 左座標、 lpPoints [1]->y : 右座標

を示す。

wCount は lpPoints配列の個数を示すが、スキャンラインが 1 本の場合は 2 , 2 本の場合は 3 , n 本の場合は n + 1 がセットされている。

【 0 0 9 4 】

図 8 は、Output関数で複数回Scanline描画指定を受けた時に、圧縮処理を行わない場合と、以上二つの手法を組み合わせた場合の動きを示す模式図である。描画領域 8 0 0 に図形 8 0 2 を描画する場合、Windows9XはOutput関数 8 0 1 を使ってY座標の小さい方から順にScanline描画を指定してくる場合がある。

【 0 0 9 5 】

一つのOutput関数に対して一つのヘッダを持つP D F 8 0 3 を作ると、同一のブラシやペンといった重複データが何度も登録されることになる。

【 0 0 9 6 】

図 8 の 8 0 4 は、第一の方法によって、同一ブラシ、ペン、描画属性はこれらの情報を一つのヘッダとして扱うことでヘッダサイズの削減が可能であることを示している。

【 0 0 9 7 】

さらに、図 8 の 8 0 2 のようにScanline同士が接している場合、以下の方法によって全座標情報を保持するよりも、データサイズを小さくすることが可能である。

【 0 0 9 8 】

まず、Scanlineデータ圧縮時には一次バッファ管理用にLASTDATA以外にも以下のメンバーを保持する。

【 0 0 9 9 】

```
typedef Struct
```

```
{
```

```
    WORD repetcnt;           // 繰り返し回数
    WORD lastY;              // 前回の Y 座標
    WORD lastLeftX;          // 前回の左 X 座標
    WORD lastRightX;         // 前回の右 X 座標
    WORD delLeftX;           // 前回の左 X 座標変化量
    WORD delRightX;          // 前回の右 X 座標変化量
```

```
} CMPSCNLines;
```

CMPSCNLines にそれまでのScanlineの情報を蓄えることにより、図 1 1 ～図 1 4 に示すコマンドを生成する。

【0 1 0 0】

次に、コマンド使用例について説明する。

【0 1 0 1】

圧縮コマンドは基本的に、〔初期座標〕〔変化分コマンド〕の連続である。

【0 1 0 2】

以下に、コマンド排出例を示す。

【0 1 0 3】

1. 縦一直線

縦一直線の場合は、「長さ分連続する変化分 0 のSCANLINE」となる。

従ってコマンドは、

〔（番号 0）（初期座標）〕

〔（番号 1 5）（連続数＝長さ）〕

〔（番号 3）（左変化分＝0）（右変化分＝0）〕

となる。

【0 1 0 4】

2. 斜め 4 5 度線

斜め 4 5 度線の場合は、左右変化分が 1 となる場合がほとんどである。

よって「長さ分連続する変化分 1 のSCANLINE」となるので、コマンドは、

[(番号 0) (初期座標)]

[(番号 1 5) (連続数=長さ)]

[(番号 3) (左変化分=1) (右変化分=1)]

となる。

【 0 1 0 5 】

3. 斜め線

一般的な斜め線では、各SCANLINE毎の変化はまちまちである。

そこで変化分が変わったその都度コマンドを発行する。

[(番号 0) (初期座標)]

[(番号 3) (左変化分) (右変化分)]

[(番号 3) (左変化分) (右変化分)]

[(番号 3) (左変化分) (右変化分)]

:

となる。

【 0 1 0 6 】

図 9 は、以上のコマンド生成処理の流れを示すフローチャートである。図 9 における各ステップの処理内容は、以下の通りである。

【 0 1 0 7 】

ステップ S 9 0 1 : 一次キャッシュ内に格納されている前回のScanlineデータの Y 座標LastYと、今回のScanlineの Y 座標の差delYを求める (一次キャッシュ内に未格納時はdelY=0)。

【 0 1 0 8 】

ステップ S 9 0 2 : delYが0以外ならステップ S 9 0 5 へ、0ならステップ S 9 0 3 へ制御を移す。

【 0 1 0 9 】

ステップ S 9 0 3 : 今回のScanlineの左XとlastLeftX, 右Xと lastRightXとの差分を求める。

【 0 1 1 0 】

ステップ S 9 0 4 : 左右とも前回の差分 delLeftX, delRightXと同じならス

テップ S 9 0 9 へ、どちらか一つでも違う場合はステップ S 9 0 5 へ制御を移す。
。

【 0 1 1 1 】

ステップ S 9 0 5 : 繰り返しカウンタ `repetcnt` が 0 ならステップ S 9 0 6 へ、それ以外ならばステップ S 9 0 7 へ制御を移す。

【 0 1 1 2 】

ステップ S 9 0 6 : 連続コマンド (`CmndNo.15`) を一次バッファ 3 0 2 4 にバッファリングし、続けて `delLeftX`, `delRightX` を示すコマンドをバッファリングし、ステップ S 9 0 7 へ制御を移す。

【 0 1 1 3 】

ステップ S 9 0 7 : 新規座標コマンド (`CmndNo.0`) で Y 座標、左右 X 座標を一次バッファ 3 0 2 4 にバッファリングし、ステップ S 9 0 8 へ制御を移す。

【 0 1 1 4 】

ステップ S 9 0 8 : `lastLeftX`, `lastRightX` にステップ S 9 0 7 でセットした左右 X 座標をセットし、`delLeftX`, `delRightX` は 0 で初期化する。

【 0 1 1 5 】

ステップ S 9 0 9 : 繰り返しカウンタ `repetcnt` をカウントアップし、ステップ S 9 1 0 へ制御を移す。

【 0 1 1 6 】

ステップ S 9 1 0 : `lastLeftX`, `LastRightX` を更新する。

【 0 1 1 7 】

なお、本発明は、複数の機器（例えばホストコンピュータ、インターフェース機器、リーダ、プリンタなど）から構成されるシステムに適用しても、1つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用しても良い。

【 0 1 1 8 】

また、本発明の目的は、前述した実施形態の機能を実現するプリンタドライバと呼ばれるソフトウェアプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（または CPU や MPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによ

っても、達成されることは言うまでもない。

【 0 1 1 9 】

この場合、記憶媒体から読み出されたプログラムコード事態が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【 0 1 2 0 】

プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM, CD-R, 磁気テープ、不揮発性のメモ리카ード、ROMなどを用いることができる。

【 0 1 2 1 】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【 0 1 2 2 】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることはいうまでもない。

【 0 1 2 3 】

本発明を記憶媒体に適応する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードを格納することになる。

【 0 1 2 4 】

本発明は、前述した実施の形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体からそのプログラムをパソコン通信など通信ラインを介して要求者にそのプログラムを配信する場合にも適用できることは言うまでもな

い。

【 0 1 2 5 】

【発明の効果】

以上説明したように、本発明によれば、プリンタドライバで中間言語を一旦スプールする場合、保持データサイズ・個数を抑えることで高速な印刷が可能になる。すなわち、同一図形をOSが別関数で描画指定をしてきた場合も、スプールオブジェクト個数、スプールデータサイズを抑えることにより、高速な印刷処理が可能となる。

【図面の簡単な説明】

【図 1】

本発明を適用したレーザビームカラープリンタ（LBP）の構造を示す側断面図である。

【図 2】

従来の描画方法を示す概念図である。

【図 3】

本発明を適用した印刷システムの全体的基本構成を示すブロック図である。

【図 4】

RECTANGLE 描画命令を個々にヘッダを付けてPDFに落とした様子を示す模式図である。

【図 5】

描画属性が同じ描画命令のヘッダを共通化する処理系を示すフローチャートである。

【図 6】

本実施の形態によって複数に分けて渡されたScanline描画命令に一つのヘッダを付けてPDFに落とした様子を示す模式図である。

【図 7】

Windows9X：DDIの一つであるOutput関数(Scanline)と、パラメータの関係を示す模式図である。

【図 8】

Windows9XのScanlineで図形を描画した様子と、従来のPDFおよび本実施の形態によるPDFの違いを示す模式図である。

【図 9】

別関数で送られてきたScanlineデータを圧縮する手順を示すフローチャートである。

【図 1 0】

従来のScanline描画命令のPDF構造体を示す図である。

【図 1 1】

本実施の形態で用いるコマンドの説明図である。

【図 1 2】

本実施の形態で用いるコマンドの説明図である。

【図 1 3】

本実施の形態で用いるコマンドの説明図である。

【図 1 4】

本実施の形態で用いるコマンドの説明図である。

【符号の説明】

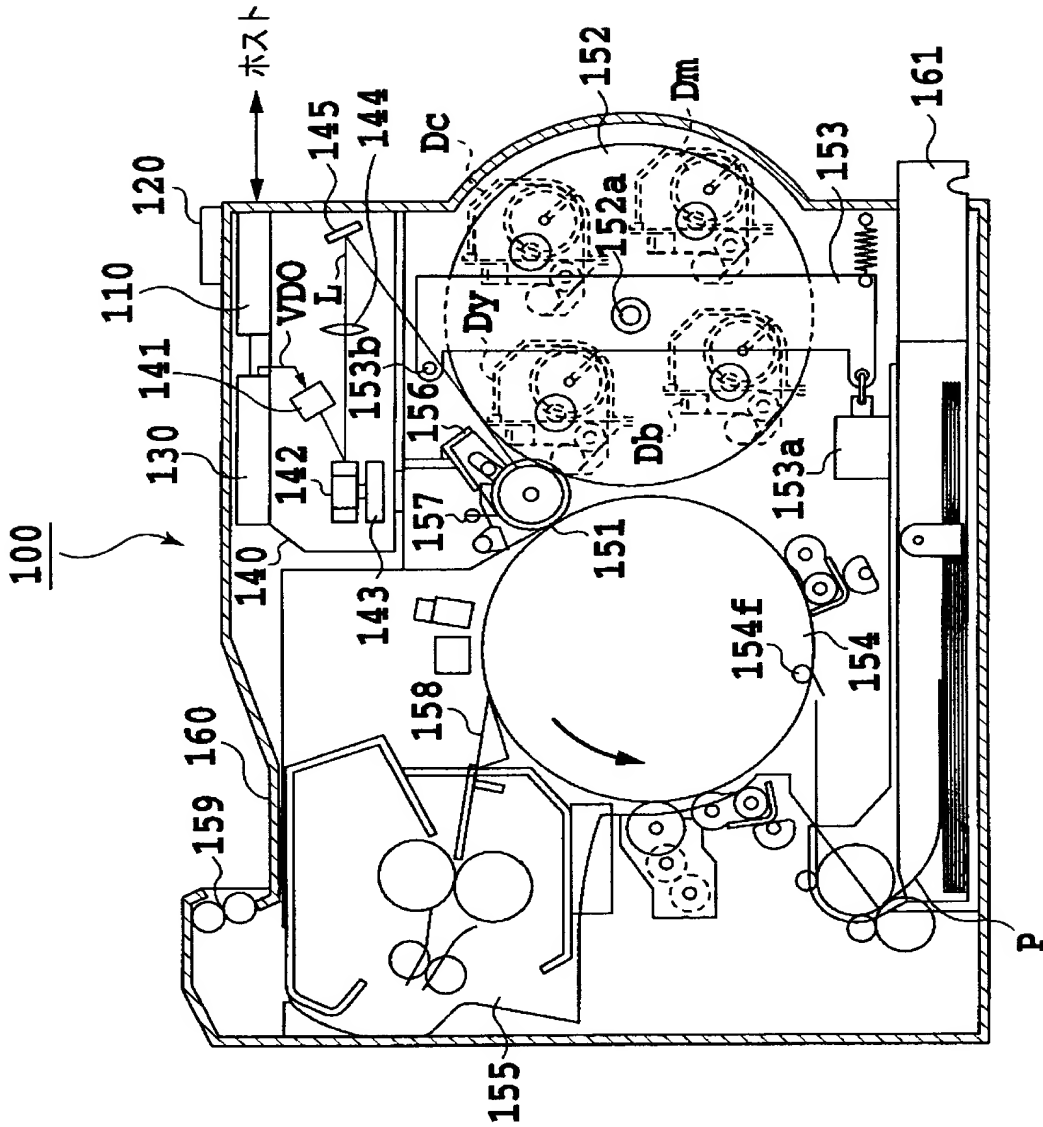
- 1 0 0 印刷装置 (L B P)
- 1 1 0 フォーマッタ制御部
- 1 1 1 インターフェース部
- 1 1 2 1 受信バッファ
- 1 1 2 2 送信バッファ
- 1 1 3 コマンド解析部
- 1 1 4 印刷制御処理実行部
- 1 1 5 描画処理実行部 (レンダラ)
- 1 1 6 ページメモリ
- 1 2 0 オペレーションパネル
- 1 3 0 出力制御部
- 1 4 0 光学ユニット
- 1 4 1 半導体レーザ

- 1 4 2 ポリゴンミラー
- 2 0 0 ホストコンピュータ
- 2 0 1 アプリケーションソフト
- 2 0 2 グラフィックサブシステム
- 2 0 2 1 G D I
- 2 0 2 2 D D I I / F
- 2 0 2 3 プリンタドライバ
- 2 1 0 キーボード
- 2 1 1 マウス
- 2 2 0 ディスプレイモニタ
- 2 0 3 スプールサブシステム
- 2 0 3 1 P D F
- 2 0 3 2 デスプーラ
- 2 0 3 3 P D L トランスレータ
- 2 0 3 4 イメージレンダラー (1 B P P)
- 2 0 3 5 イメージレンダラー (Y M C K 4 B P P)
- 2 0 3 6 イメージレンダラー (R G B 2 4 B P P)
- 2 0 3 7 バンドメモリ
- 2 0 3 8 I / F 部
- 3 0 0 ホストコンピュータ
- 3 0 1 アプリケーションソフト
- 3 0 2 グラフィックサブシステム
- 3 0 2 1 G D I
- 3 0 2 2 D D I I / F
- 3 0 2 3 プリンタドライバ
- 3 0 2 4 一次バッファ
- 3 0 3 スプールサブシステム
- 3 0 3 1 P D F (プリントデータファイル)
- 3 0 3 2 デスプーラ

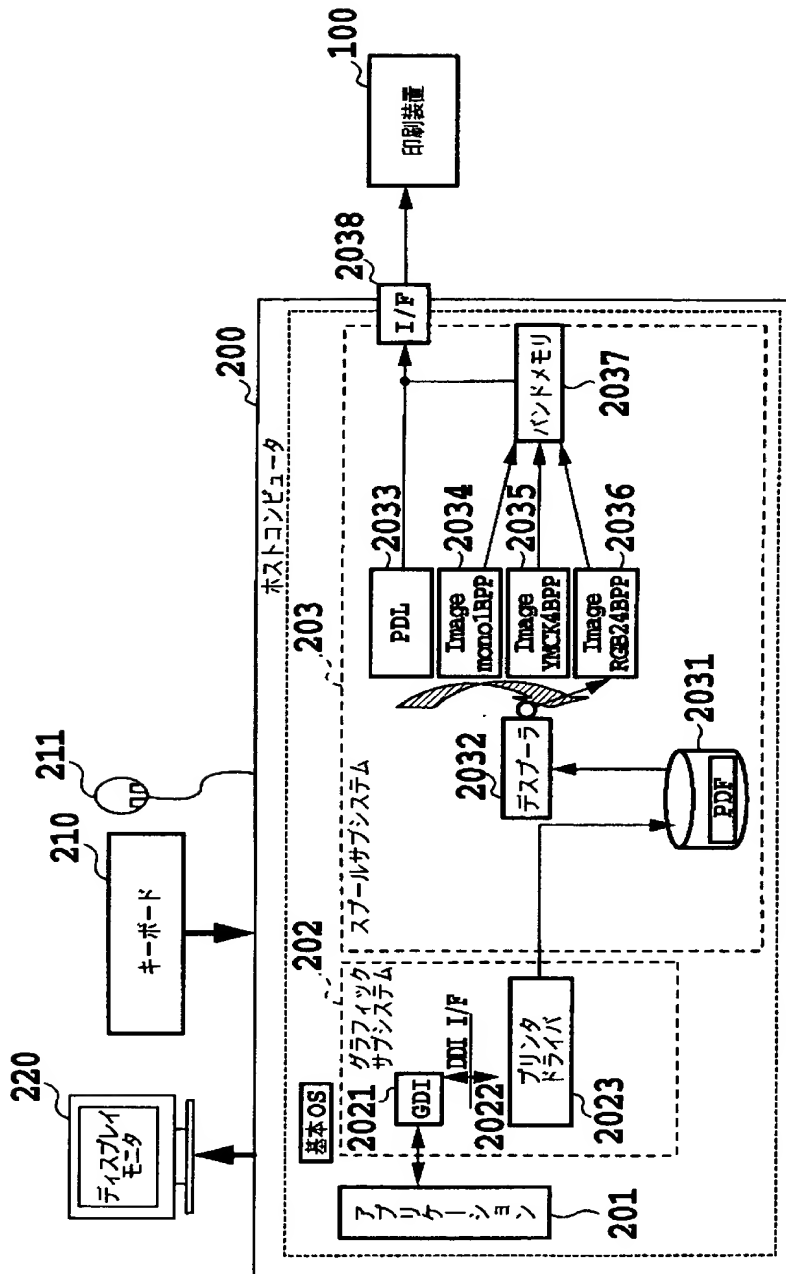
- 3 0 3 3 P D L ト ラ ン ス レ ー タ
- 3 0 3 4 モ ノ ク ロ 1 B P P イ メ ー ジ レ ン ダ ラ ー
- 3 0 3 5 Y M C K 4 B P P イ メ ー ジ レ ン ダ ラ ー
- 3 0 3 6 R G B 2 4 B P P イ メ ー ジ レ ン ダ ラ ー
- 3 0 3 7 イ メ ー ジ モ ー ド 用 バ ン ド メ モ リ 領 域
- 3 0 3 8 イ ン タ ー フ ェ ー ス 部

【書類名】 図面

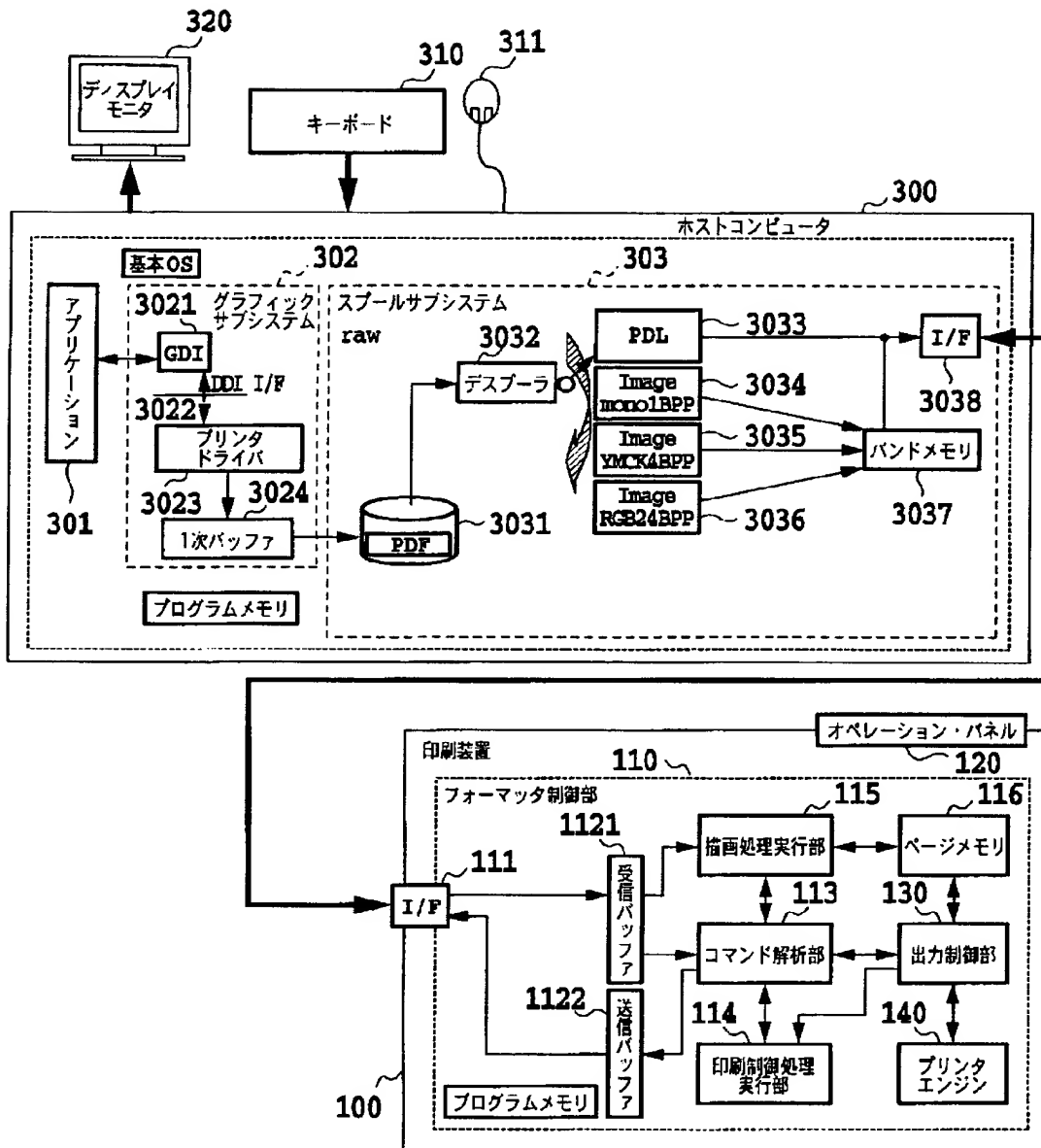
【図 1】



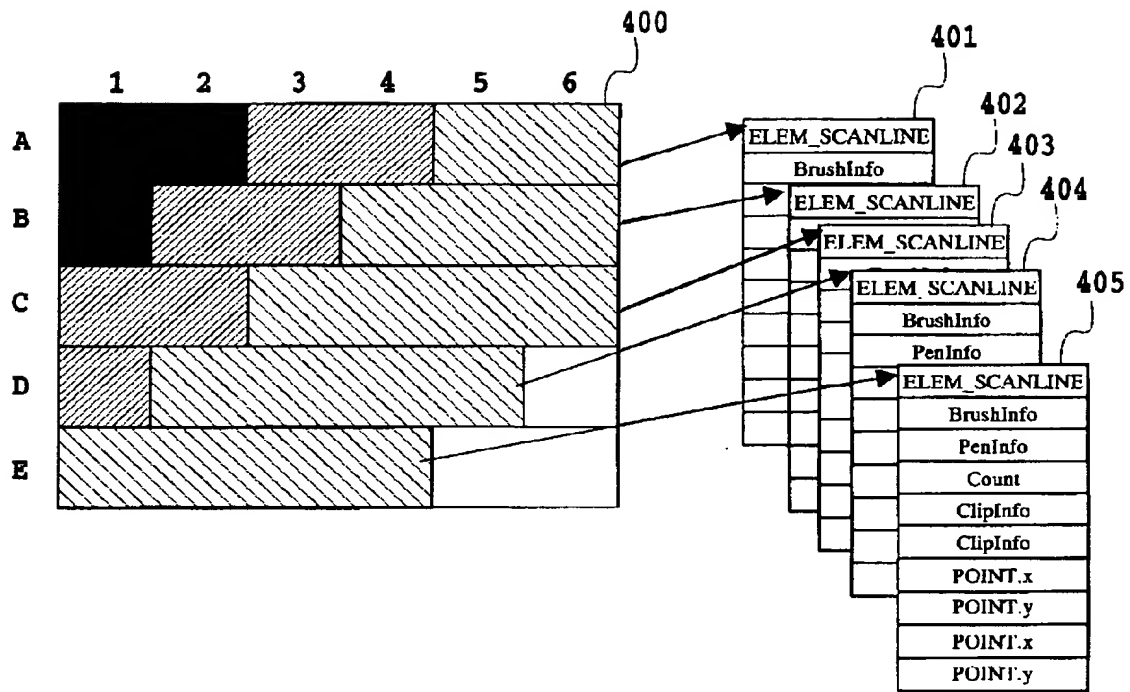
【図 2】



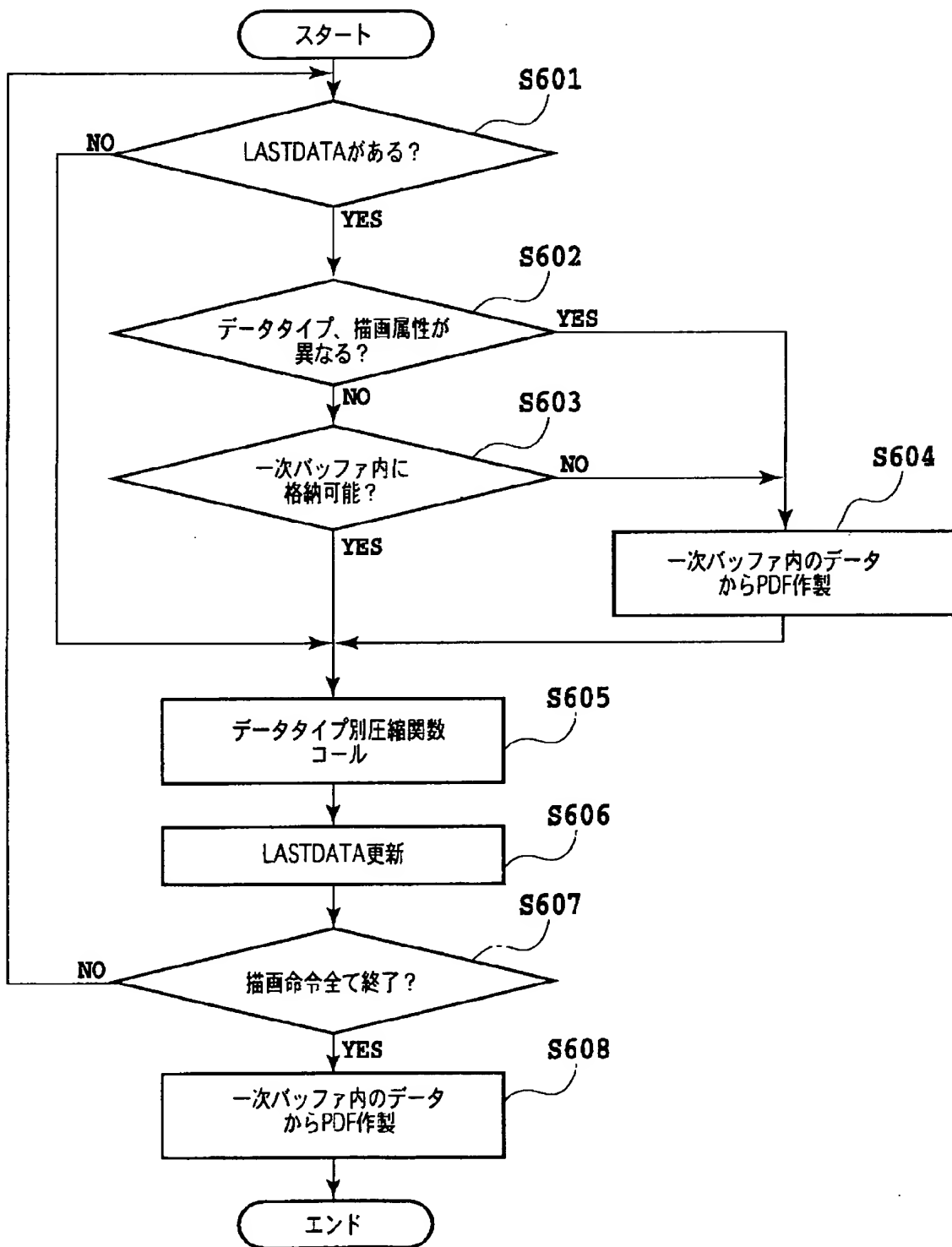
【図 3】



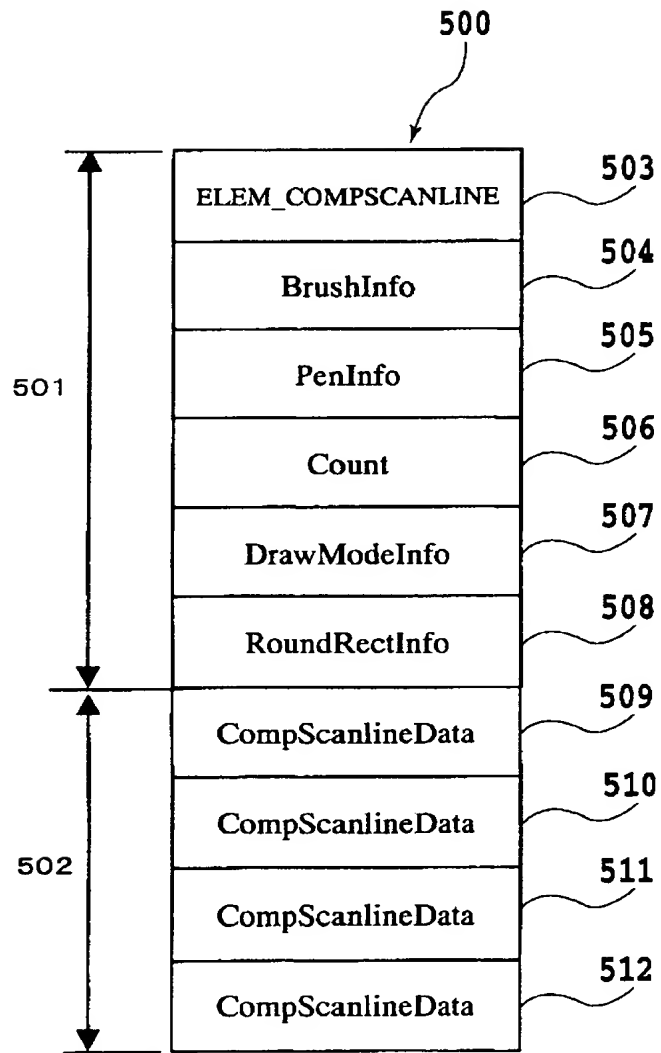
【図 4】



【図 5】



【図 6】



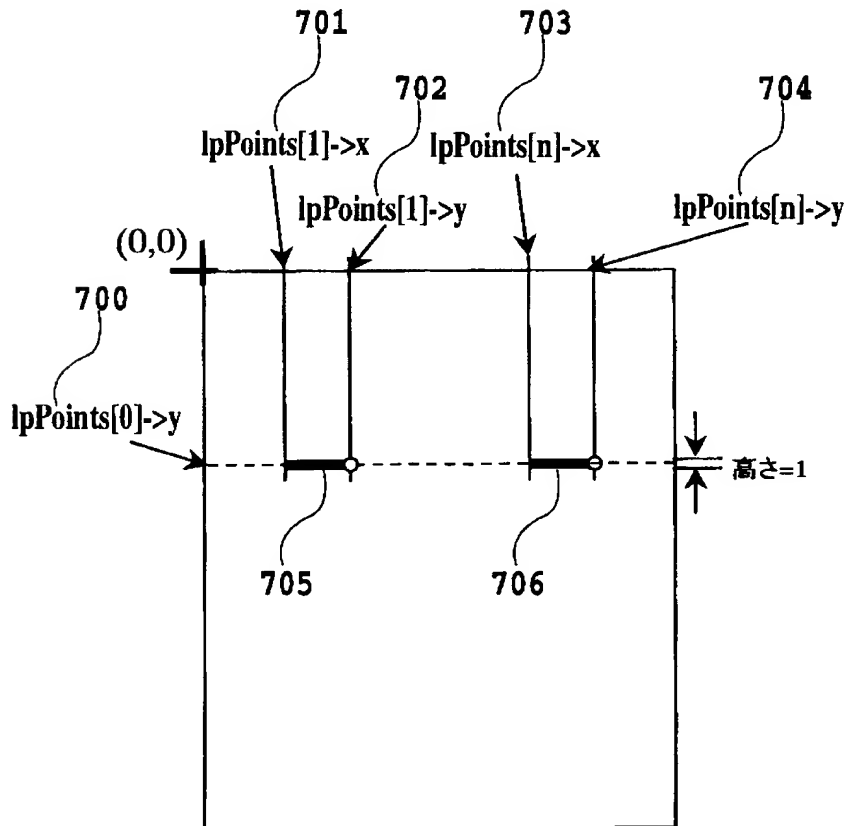
【図 7】

関数名:

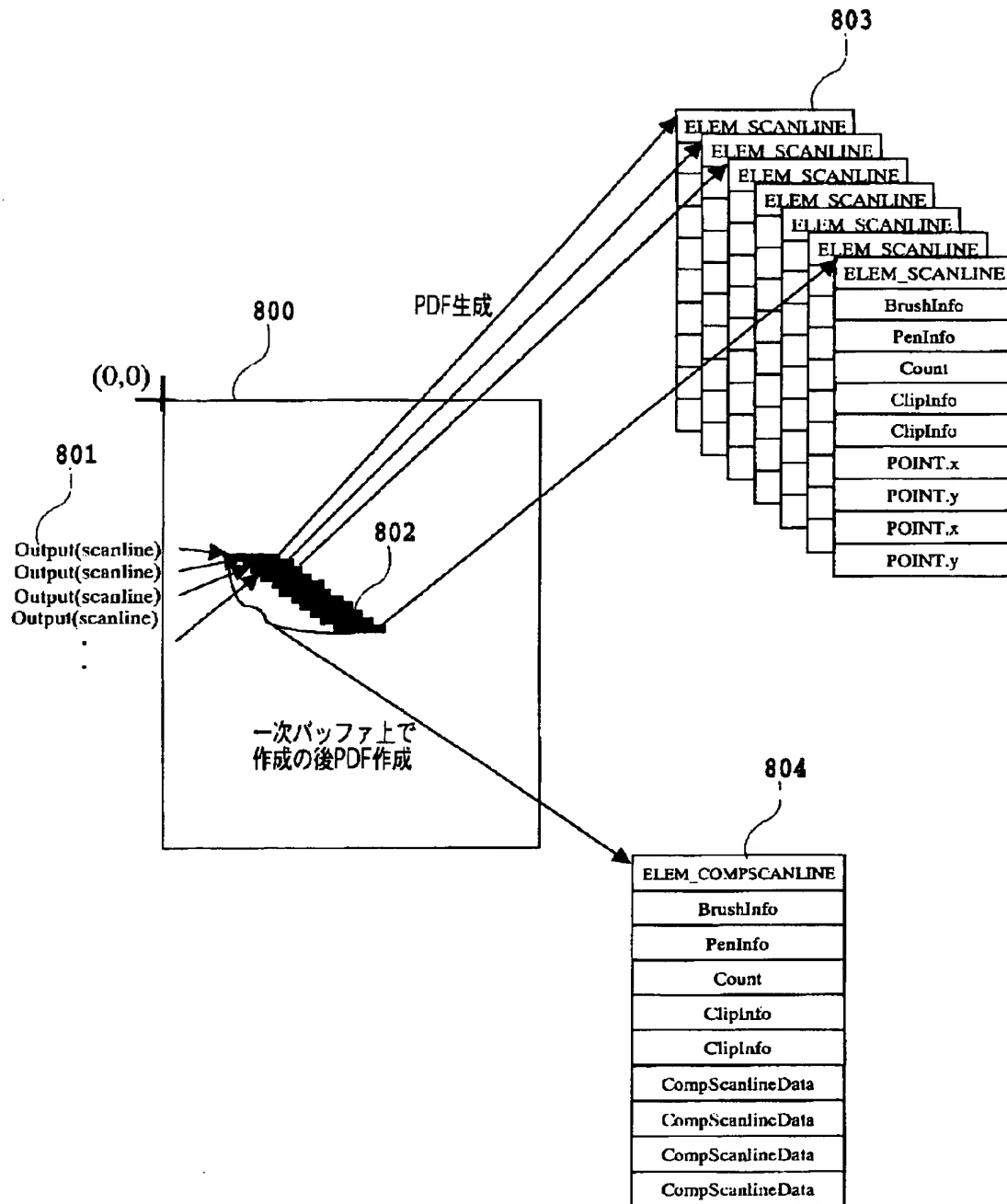
Output(LPPDEVICE lpDestDev, WORD wStyle,
WORD wCount, LPPOINT lpPoints,
LPPEN lpPPen, LPPBRUSH lpPBrush,
LPDRAWMODE lpDrawMode,
LPRECT lpClipRect)

引数

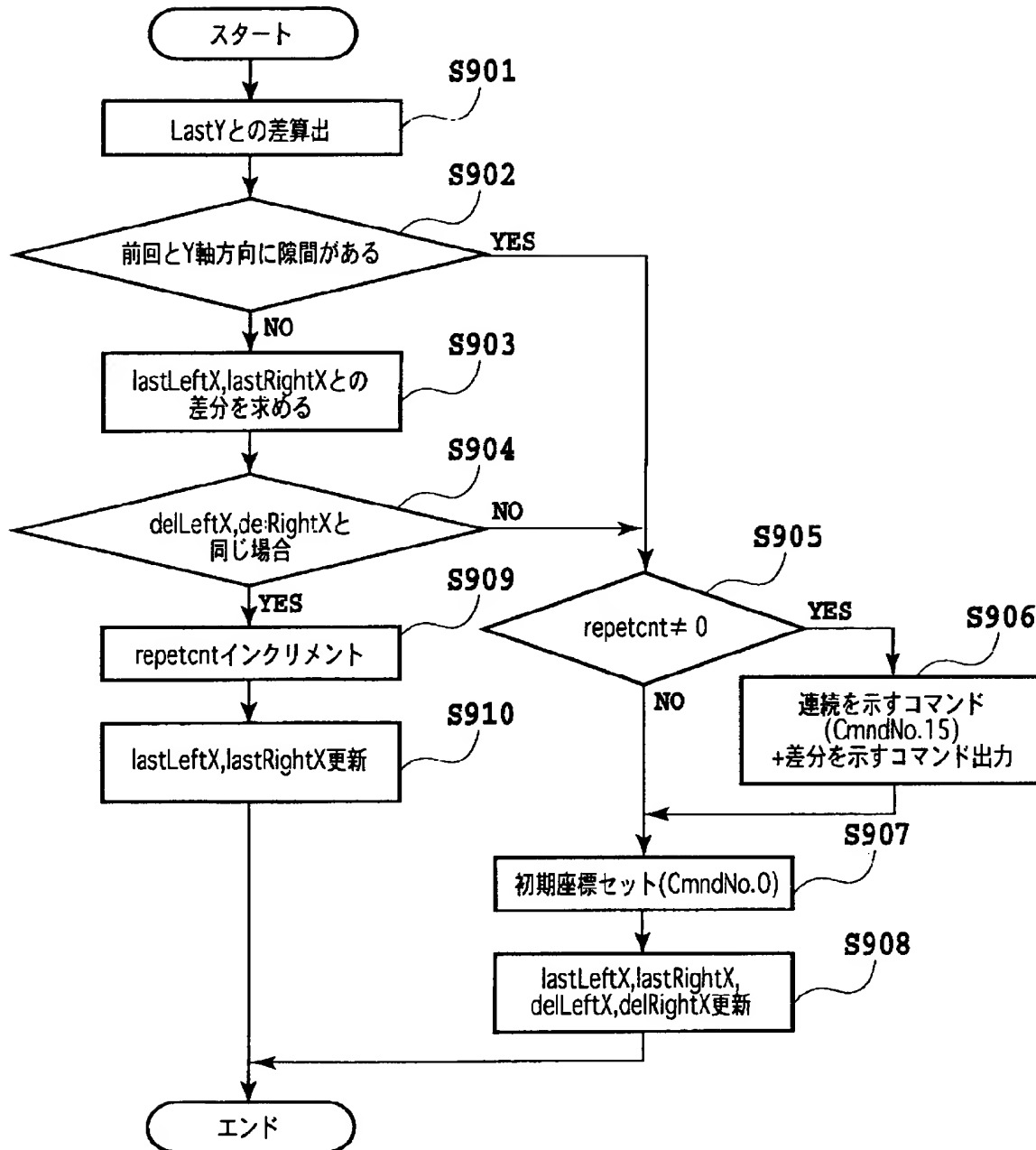
LPPDEVICE	lpDestDev	PDEVICE or PBITMAP
WORD	wStyle	OS_SCANLINES
WORD	wCount	lpPoints引数の点数
LPPOINT	lpPoints	スキャンラインの始点-終点の配列
LPPEN	lpPPen	物理ペン
LPPBRUSH	lpPBrush	物理ブラシ
LPDRAWMODE	lpDrawMode	描画コンテキスト
LPRECT	lpClipRect	クリップ矩形



【図 8】

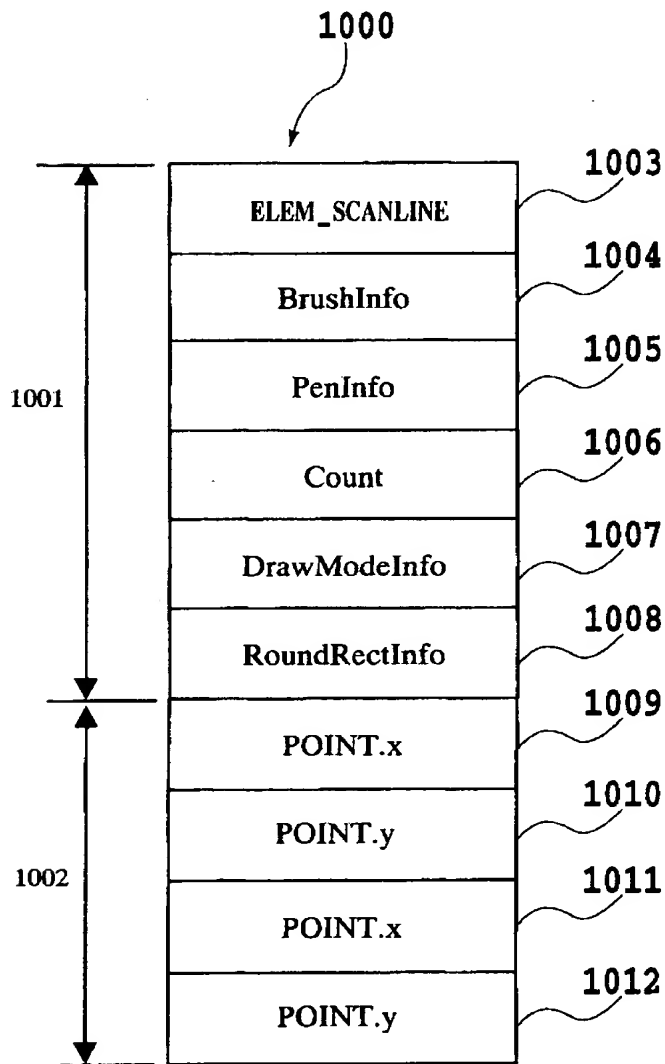


【図 9】



【図 1 0】

PDFの構造を示す模式図

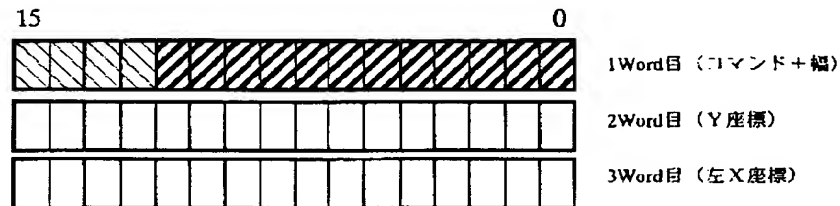


【図 1 1】

(コマンドデータ構成)

コマンドとは、COMPScanlineデータ部の最上位4ビット部を使い、圧縮したデータ識別用に設けられたコード(0～12)を示し、その値により圧縮されたデータ構造が異なる。

・コマンドコード 0 (3Word 構成)



Y座標、左X座標はサイン付パラメータを示す。

※サイン付とは、最上位ビットが符号で、他のビットは値を示す。

(0ビット以外は絶対値である)

【図 1 2】

・コマンドコード 1、2 (1Word 構成)

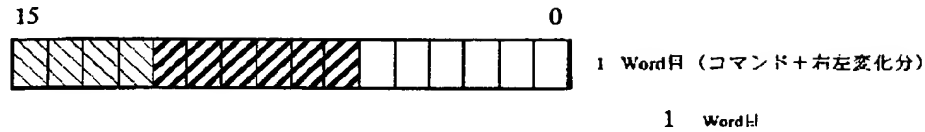


コマンドコード 1は、前回の座標データと今回の座標データを比較し、Y座標変化量が+1、left側座標は前回と同じ、かつright側座標変化量を11ビットで表現可能な変化範囲の場合に使用される。(ビット長は12だが、1ビットは符号で使用)

同様にコマンドコード 2は、Y座標変化量が+1、right側座標が前回と同じ、かつleft側座標変化量を11ビットで表現可能な場合に使われる。

【図 1 3】

・ コマンドコード 3 (1Word 構成)



上記コマンドコード 3は、前回の座標データと今回の座標データを比較し、Y座標変化量が+1、leftおよびRight座標変化が5ビットで表現可能な変化範囲の場合に使用される。

(ビット長は6だが、1ビットは符号で使用)

・ コマンドコード 4、5 (1Word 構成)



コマンドコード 4は、前回の座標データと今回の座標データを比較し、Y座標変化量が-1、left側座標は前回と同じ、かつright側座標変化量を11ビットで表現可能な変化範囲の場合に使用される。(ビット長は12だが、1ビットは符号で使用)

同様にコマンドコード 5は、Y座標変化量が+1、right側座標が前回と同じ、かつleft側座標変化量を11ビットで表現可能な場合に使用される。

【図 1 4】

・ コマンドコード 6 (1Word 構成)



上記コマンドコード 6は、前回の座標データと今回の座標データを比較し、Y座標変化量が-1、leftおよびRight座標変化が5ビットで表現可能な変化範囲の場合に使用される。

(ビット長は6だが、1ビットは符号で使用)

・ コマンドコード 15 (1Word 構成)



上記コマンドコード 15は、前回と今回の座標データを比較することで求められるコマンドコード番号が前回のものと同じ場合、その連続数を12ビットで表現できるときに使用される。尚、12ビット以上になるときは、コマンド0を使用して初期座標の再設定を行う。

【書類名】 要約書

【要約】

【課題】 プリンタドライバで中間言語を一旦スプールする場合、保持データサイズ・個数を抑えることにより高速な印刷を可能にする。

【解決手段】 プリンタドライバが D D I 関数を介して受け取った描画命令の情報を、描画オブジェクトタイプ毎のタグをつけて、描画位置情報、色情報、パターン情報、描画論理情報といった属性と共に、一次バッファ 3 0 2 4 に保持する。一次バッファ 3 0 2 4 に保持する際に、既に保持されている描画オブジェクトがある場合、タイプ、描画属性が同じ時は一次バッファ 3 0 2 4 の中に追加登録していく。「タイプが異なる。」「一次バッファにデータが入りきらない」、「描画属性が変化する」、「ページ処理終了」の何れかに当てはまるときは、既に一次バッファ 3 0 2 4 に入っている描画オブジェクトから P D F 3 0 3 1（ページデータファイル）を生成し、スプールする。

【選択図】 図 3

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 1 0 0 7]

1. 変更年月日 1 9 9 0 年 8 月 3 0 日
[変更理由] 新規登録
住 所 東京都大田区下丸子 3 丁目 3 0 番 2 号
氏 名 キヤノン株式会社